Interpreter for the deployment of intelligent tutoring systems in mobile devices

Ramón Zatarain-Cabada, M. L. Barrón-Estrada, Jesus E. Urías Barrientos

Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n, Col. Guadalupe, Culiacán Sinaloa, 80220, México.

(Paper received on September 07, 2010, accepted on October 20, 2010)

Abstract. The main objective of this paper is to propose an application to display educational content in mobile devices, adaptable to the way that each user learns, with the added value of allowing students to carry out the learning process at any time and any place. The educational content is tailored to the student's learning style according to the model of Felder-Silverman. The identification of the student's learning style is performed using self-organizing maps. The operation of the tool presented, is based on techniques such as parsing XML files and deployment of user interfaces on mobile devices. It also makes use of neuron networks, specifically, self-organizing maps or Kohonen maps to perform detection of learning style of students.

Keywords: Adaptive mobile learning, Authoring tools, Learning Styles.

1 Introduction

Shaping the study materials, according to the characteristics of the student, on mobile devices is a topic of current research. However, most research on mobile intelligent tutoring systems (MITS, for its acronym in English) is more suited to systems that do not support artificial intelligence techniques to develop the adaptive implementation [1, 2, 3]. This is largely because mobile devices have many limitations, such as its low processing power, small memory and low speed at which access it. In addition, we also found that the user interface is rather limited (the screens are small and the keyboards have only a few keys). This causes that the information displayed to the user, is limited to some few lines of text and perhaps some images at once. Because of these limitations, any application designed to run on mobile devices should make a wise use of available resources. For example, the size of the files of the application must be as short as possible, as space is quite limited. In addition, processes should not be too demanding of the processor (should not take many clock cycles), the memory used should be the minimum possible, and the information that must be displayed to the user must be brief and concise.

Here we propose a visualization tool for intelligent tutoring systems, built on the platform EDUCA [4], which lets you create Intelligent Tutoring Systems (ITS) using the theory of learning styles of Richard Felder and Linda Silverman [5]. These ITS are stored using XML files, which allows the interpretation to be performed easily and keeps the cost of processing and memory usage at a low level. To make the adaptation to the student's learning style, the tool uses a self organizing map, which was designed with the goal of requiring few resources, without compromising its effectiveness, making it ideal to run on mobile devices.

The paper's organization is as follows: In Section 2, we present the layer architecture of the interpreter. In Section 3, we explain how an ITS is built. Section 4 gives a short introduction of the Predictive Engine. Results and Discussions are given in Section 5. Conclusions and future work are discussed in Section 6.

2 Interpreter Layer Architecture

The architecture of the interpreter is shown in Figure 1. We have placed only the most significant components of the whole application.

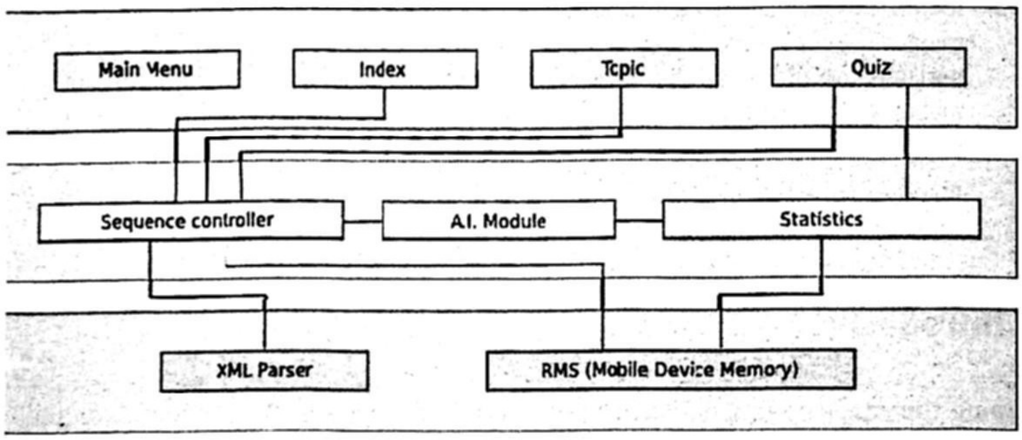

Figure 1. *Mobile interpreter architecture*

From all the components, **MainMenu** allows the first contact with the user since it is connected with **Sequence Çontroller**, which selects and shows the contents to the user.

In addition, **Quiz** is related with **Statistics** (business layer) which in turn is related with **RMS** (the data persistence module) in the data layer, to store the results of the evaluation.

On the other hand, component **AI module** has a link to **Sequence controller** and to **Statistics**; both of them provide the intelligent module with information about the user interaction with the course; with that information the AI module performs several actions (using the neural network) to determine which information is shown and how it is going to be shown to the user according to the learning style.

The interpreter was implemented using Java MicroEdition and XML since both technologies are platform independent. Java has been chosen as the language for the mobile application to take advantage of the multiplatform support, which helps the application to work on a wide range of mobiles without requiring neither complete nor partial reprogramming.

3 Building the ITS

The process of constructing an intelligent system consists of three main steps (figure 2). During Step 1 a tree structure of the adaptive or intelligent tutoring system is designed by the main instructor(s) of the learning material. On the tree structure, the designer also inserts quizzes (multiple selection and choice). Quizzes are an important element to provide adaptation capabilities to the produced tutors.

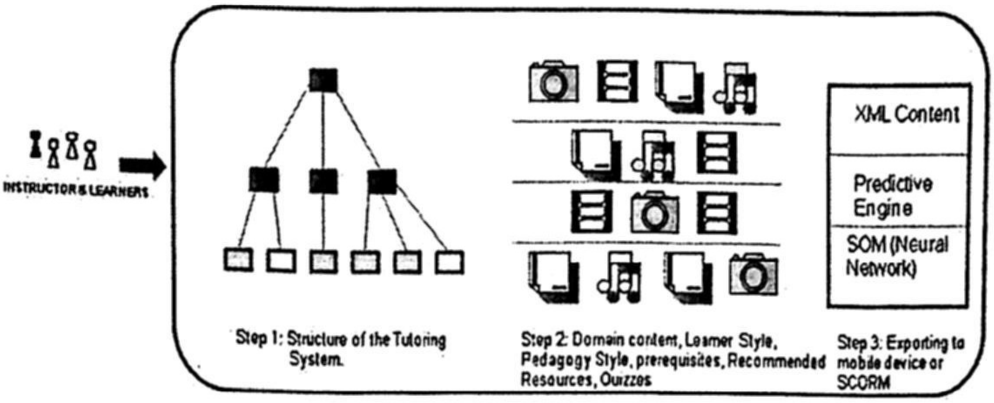

Figure 2. *Building a Tutoring System*

In step 2 the tree structure is filled with domain contents (a knowledge base), and some other learning resources. At the beginning of the creation the instructor or teacher authors the tutoring system by inserting different learning objects like text fragments, images, audio/video clips, and by defining learning styles, prerequisites, tags and quizzes. At a later time, learning resources are added to the tutoring system by learners, who recommend resources they find commonly on the web. The third step consists of saving/exporting a package with the learning resources or contents (an XML file), a predictive engine for navigation purposes, and a SOM Neural Network for learning style classification.

3.1 Creating the Knowledge Base

In order to design and implement a course knowledge representation, we apply some of the concepts related to Knowledge Space Theory [6]. This theory provides a sound foundation for structuring and representing the knowledge domain for personalized or adaptive learning. An adaptive assessment based upon knowledge spaces and student prerequisites (and employing a Kohonen neural network [7] for identifying learning styles), will derive a particular or personalized path of learning objects. For each topic a set of prerequisites and quizzes are set. Figure 3 shows the knowledge domain of the topic *parsing* or syntactic analysis for a compiler construction course. Dashed lines represent those paths. We can also see the prerequisites for studying the topic (Lecture 2 – Lexical Analysis).

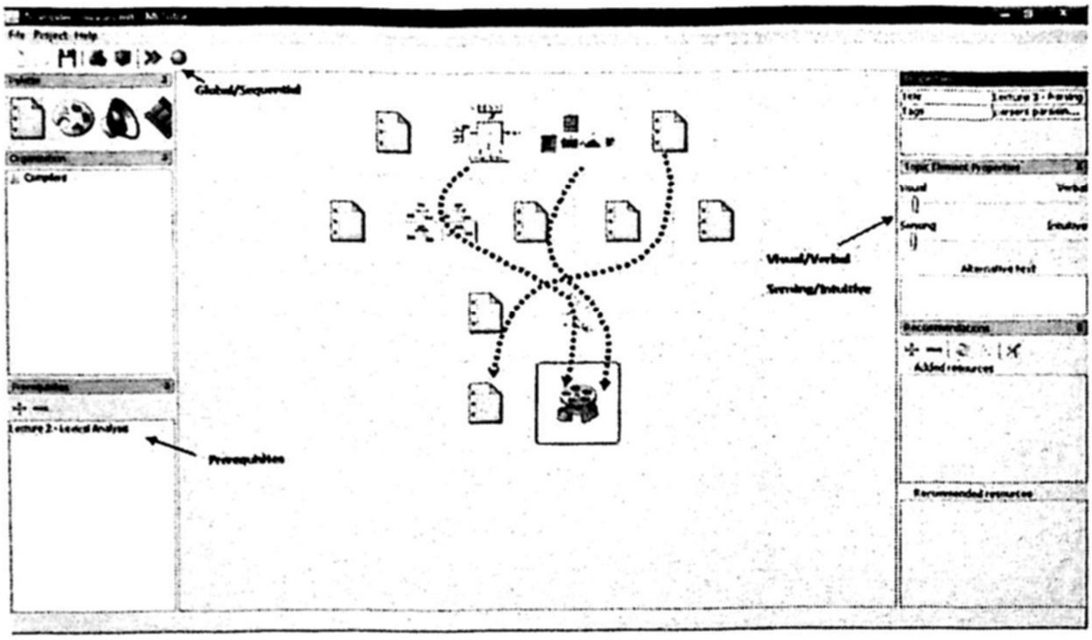

Figure 3. Knowledge Domain for the parsing topic

3.2 Implementing the ITS

The storage of the contents of the course is done using XML files. To reduce the workload of the mobile device, the information is divided into multiple files, that when the course is exported, this is inserted into the mobile application. Next, part of an xml file for a tutoring system is shown.

```
<?xml version="1.0" encoding="UTF-8"?>
  <topic>
        <title>Graficas</title>
        <resources>
              <resource>http://www.cucei.udg.mx/.../histograma.html</resource>
              <resource>http://www.hrc.es/bioest/Ejemplos_histo.html</resource>
        </resources>
        <prerequisites />
        <row rank="0">
              <component>
                    <sensing_intuitive>20</sensing_intuitive>
                    <visual_verbal>100</visual_verbal>
                    <alt></alt>
                    <sequential type="text">
                          <source>text0.xml</source>
                    </sequential>
                    ...
                    <global type="image">
                          <source>-832547520.png</source>
                    </global>
              </component>
              ...
        </row>
        ...
  </topic>
```

Each of the themes of an ITS is represented with file *topic#.xml* where # is the identifier of the item. The elements of the file are:

- Topic. File root element which has components *title, resources, prerequisites* and *row* as children.
  - Title: The name of the topic.
  - Resources: the description and URL of a resource.
  - Prerequisites: Topics the students need to know to understand the current topic.
  - Row: Within each file *topic#.xml*, it must be one or more row elements, which represent a learning unit. Each *row* element contains one or more *component* elements. *Component* is part of the learning unit. It contains elements that define the degree of belonging to this component, to Felder-Silverman dimensions "visual / verbal", "sensitive / intuitive", and "sequential / global".

4 The Predictive Engine

The main goal of the predictive engine is to dynamically identify the student's learning style whenever he/she is running the tutoring system. At the beginning, an interpreter selects content (learning objects) based upon the student's learning style, obtained from a student profile created since the first session of the student. The learning style can be modified according to evaluations applied to the student.

4.1 The Self-Organizing Map

The selection of a neural network model for solving a problem depends on its nature of it. The choice of using self-organizing maps was motivated mainly by three factors: the unsupervised type of learning of such networks, the network performance and the training speed. In supervised learning, we assume the existence of an expert on the problem that we are trying to solve with the neural network. In unsupervised learning, neural networks keep relationships between the input signals independently. Identifying learning styles of students is a complex task which falls in the discipline of pedagogy. The proper training of a neural network with supervised learning requires, in most cases, a pedagogue. With self-organizing maps we do not need a specialist in the training process.

5 Results and Discussions

The interpreter was implemented first in desktop environments and mobile devices. The desktop environment was used to edit the intelligent tutoring systems while the mobiles were used for deployment of them. Currently the interpreter has been implemented also in a Web 2.0 social network. This learning social network has the name of Zamná [8]. Figure 4 shows a deployment of an ITS for a compiler course in both environments (The web and Mobile devices). One of the advantages of using a social learning networking is the exchanging and sharing of learning objects among community members.

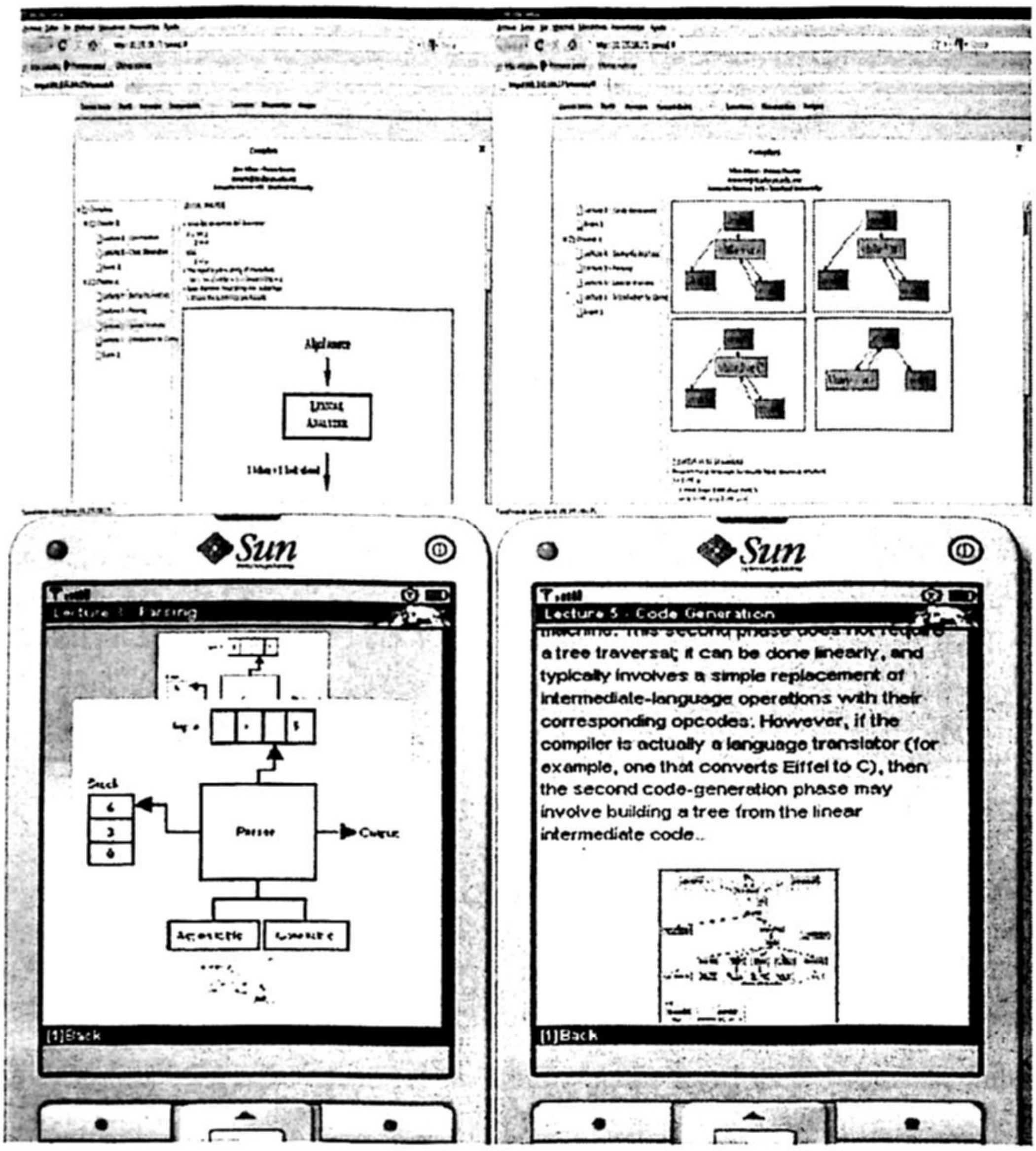

Figure. 4. *The Compiler Course on the Web and Mobile Devices*

6 Conclusions and Future Work

The interpreter for mobile devices is a good tool to support education "anywhere, anytime", because it allows to carry out the learning task using a device that most people have, and carries almost any moment, this, without sacrificing quality of content and also adding the ability to tailor the information displayed to the way each student learns.

Currently, the interpreter is totally being migrated to the Web 2.0 environment of Zamna. For future work, we want to export tutoring systems to other mobile platforms like Microsoft .NET Compact Framework [9], and the iPhone OS [10].

Acknowledgments.

The work described in this paper is fully supported by a grant from the DGEST (Dirección General de Educación Superior Tecnológica) in México.

References

1. Bull, S.; Cui, Y.; McEnvoy, A. T.; Reid, E. y Yang, W. "Roles for Mobile Learners Models". Proceedings of the 2nd IEEE Workshop on Wireless and Mobile Technologies in Education (2004).
2. Shih, K-P.; Chang, C-Y.; Chen, H-C. y Wang. S-S. "A Self-Regulated Learning System with Scaffolding Support for Self-Regulated e/m-Learning". Proceedings of the 3rd International Conference on Information Technology: Research and Education (2005).
3. Romero, C.; Ventura, S.; Hervás, C., y De Bra, P. "An Authoring Tool for Building Both Mobile Adaptable Tests and Web-Based Adaptive or Classic Tests". Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (2006).
4. Ramon Zatarain-Cabada, Maria Lucia Barrón-Estrada, Guillermo A. Sandoval-Sánchez, Eduardo Urías Barrientos, J. Moisés Osorio-Velásquez, Carlos A. Reyes García: EDUCA: A Web 2.0 Collaborative, Mobile and E-learning Authoring System. ICALT 2009: 287-289.
5. R.M. Felder, L.K. Silverman. Learning and Teaching Styles in Engineering Education, Engineering Education, 78, pp.674-681 (1988).
6. Doignon, J. –P. and Falmagne, J. C. (1999). Knowledge Spaces. Springer-Verlag.
7. Kohonen, T. (1989). Self-Organization and Associative memory (3rd ed.). Springer-Verlag.
8. Ramon Zatarain-Cabada, Maria Lucia Barrón-Estrada, Viridiana Ponce Angulo, Adán José García, Carlos A. Reyes García: Identification of Felder-Silverman Learning Styles with a Supervised Neural Network. ICIC (2) 2010: 479-486.
9. .NET Compact Framework Developer Center. http://msdn.microsoft.com/enus/. Consultado en Septiembre de 2009.
10. iPhone Developer Center. http://developer.apple.com/iphone/. Consultado en Septiembre de 2009.